1. Method of structurally converting a binary sequence into an encrypted final image G, the structural conversion comprising the steps of:

forming an image M of the binary sequence as a concatenation of a tag data element T and structural data element S, tag data element T comprising information necessary to reverse a conversion process, structural data element S comprising a sequence of logical scales of position coding;

selecting a number of conversion function iterations P to be performed;

iteratively executing P times a conversion function comprised of the following steps:

selecting a transformation algorithm A from a predefined set of transformation algorithms L;

selecting an alphabet of transformation AV based upon the structural data element S;

applying algorithm A and alphabet AV to structural data element S to form a plurality of logical scales of position coding;

forming a transformed structural data element S' comprised of a sequence of the logical scales of position coding;

selecting an external key K^x;

forming tag data element T;

coding the tag data element T with external key K^x to obtain coded tag data element T'';

repeating the steps of the conversion function upon a converted image M' comprised of a concatenation of the coded tag data element T'' and the transformed structural data element S';

and forming the encrypted final image G as a concatenation of the coded tag data element T'' and the transformed structural data element S' created upon the P^{th}

iteration of the conversion function.

2. Method of structurally converting a binary sequence into an encrypted final image *G*, said structural conversion comprising the steps of:

forming an image *M* of the binary sequence as a concatenation of a tag data element *T* and structural data element *S*, tag data element *T* comprising information necessary to reverse the conversion process, structural data element *S* comprising a sequence of logical scales of position coding;

selecting a number of conversion function iterations *P* to be performed;

iteratively executing *P* times a conversion function comprised of the following steps:

selecting a transformation algorithm *A* from a predefined set of transformation algorithms *L*;

selecting an alphabet of transformation *AV* based upon the structural data element *S*;

applying algorithm *A* and alphabet *AV* to structural data element *S* to form a plurality of logical scales of position coding;

forming a transformed structural data element *S'* comprised of a sequence of the logical scales of position coding;

stochastically selecting a bit length parameter and a shift parameter which define an internal identifier *K* within transformed structural data element *S'*;

forming tag data element *T*;

coding a portion of the tag data element *T* with internal identifier *K* to obtain a partially coded tag data element *T'*;

selecting an external key K^x;

coding the partially coded tag data element *T'* with external key K^x to

obtain coded tag data element T'';

determining whether to extract internal identifier K from transformed structural data element S', and if determined necessary, extracting the internal identifier K from transformed structural data element S' to obtain structural data element S'' and storing internal identifier K in a file of internal identifiers *FID*;

repeating the steps of the conversion function upon a converted image M' comprised of a concatenation of the coded tag data element T'' and either transformed structural data element S' if internal identifier K was not extracted, or structural data element S'' if internal identifier K was extracted;

and forming the encrypted final image G as a concatenation of the coded tag data element T'' and either transformed structural data element S' if internal identifier K was not extracted, or structural data element S'' if internal identifier K was extracted.

3. Method of structurally converting a binary sequence into an encrypted final image G, said structural conversion comprising the steps of:

forming an image M of the binary sequence as a concatenation of a tag data element T and structural data element S, tag data element T comprising information necessary to reverse the conversion process, structural data element S comprising a sequence of logical scales of position coding;

selecting a number of conversion function iterations P to be performed;

iteratively executing P times a conversion function comprised of the following steps:

selecting a transformation algorithm A from a predefined set of

transformation algorithms L;

selecting an alphabet of transformation AV based upon the structural data element S;

applying algorithm A and alphabet AV to structural data element S to form a plurality of logical scales of position coding;

forming a transformed structural data element S' comprised of a sequence of the logical scales of position coding;

stochastically selecting a bit length parameter and a shift parameter which define an internal identifier K within transformed structural data element S';

scrambling internal identifier K with a scrambling function to obtain a scrambled internal identifier K';

forming tag data element T;

coding a portion of the tag data element T with scrambled internal identifier K' to obtain a partially coded tag data element T';

selecting an external key K^x;

coding the partially coded tag data element T' with external key K^x to obtain coded tag data element T'';

determining whether to extract internal identifier K from transformed structural data element S', and if determined necessary, extracting the internal identifier K from transformed structural data element S' to obtain structural data element S'' and storing scrambled internal identifier K' in a file of internal identifiers *FID*;

repeating the steps of the conversion function upon a converted image M' comprised of a concatenation of the coded tag data element T'' and either transformed structural data element S' if internal identifier K was not extracted, or structural data element S'' if internal identifier K was extracted;

and forming the encrypted final image G as a concatenation of the coded tag data element T'' and either transformed structural data element S' if internal identifier K was not extracted, or structural data element S'' if internal identifier K was extracted.

4. The method of claim 2, further comprising the steps of:

structurally converting the file of internal identifiers *FID* to obtain a converted file of internal identifiers *FID'*, wherein a tag data element formed during the structural conversion of the file of internal identifiers *FID* is coded with an external key selected stochastically from a multitude of external keys in an external key file K_{EXT}; and

optionally transmitting the encrypted final image G and structurally converted file of internal identifiers *FID'* to a subscriber or receiver.

5. The method of claim 1, wherein the external key K^x is selected from a multitude of external keys in an external key file K_{EXT}.

6. The method of claim 1, wherein the selection of external key K^x is a stochastic selection.

7. The method of claim 1, wherein a same external key K^x is selected for use in all iterations.

8. The method of claim 1, wherein a different external key K^x is selected upon each iteration.

9. The method of claim 1, wherein the external key K^x is entered by a user during the conversion and reverse conversion process.

10. The method of claim 5, further comprising the steps of:

structurally converting the external key file K_{EXT} to obtain a structurally converted external key file; and

transmitting to a subscriber the structurally converted external key file and an initial key K_{INIT} required to reverse the structural conversion of the structurally converted external key file to obtain the external key file K_{EXT}.

11. The method of claim 1, wherein the selection of transformation algorithm A may be a stochastic selection.

12. The method of claim 1, wherein the selection of transformation algorithm A may depend upon adherence to a mathematical criterion.

13. The method of claim 1, wherein the selection of transformation algorithm A may depend upon adherence to a logical criterion.

14. The method of claim 1, wherein the selection of transformation algorithm A may depend upon adherence to a file size criteria for encrypted final image G.

15. The method of claim 1, wherein the predefined set of transformation algorithms L may be supplemented.

16. The method of claim 1, wherein the selection of a number of conversion steps P may be a stochastic selection.

17. The method of claim 1, wherein the selection of a number of conversion steps P may depend upon adherence to a mathematical criterion.

18. The method of claim 1, wherein the selection of a number of conversion steps P may depend upon adherence to a logical criterion.

19. The method of claim 1, wherein the selection of a number of conversion steps P may depend upon adherence to a file size criteria for encrypted final image G.

20. The method of claim 1, wherein the alphabet of transformation AV is comprised of letters or quants, each letter or quant comprising a segment of structural data element S.

21. The method of claim 2, further comprising the step of determining upon which iterations, if any, internal identifiers are to be extracted.

22. The method of claim 3, further comprising the step of determining upon which iterations, if any, internal identifiers are to be extracted.

23. The method of claim 20, wherein a number of bits in each letter or quant is stochastically selected.

24. The method of claim 20, wherein a number of bits in each letter or quant may depend upon adherence to a mathematical criterion.

25. The method of claim 20, wherein a number of bits in each letter or quant may depend upon adherence to a logical criterion.

26. The method of claim 20, wherein a number of bits in each letter or quant may depend upon adherence to a file size criteria for encrypted final image G.

27. The method of claim 1, wherein the information necessary to reverse the conversion process stored in tag data element T may comprise one or more of the following:

an indicator of whether a current iterative step is the P^{th} iteration;

an indicator of whether the selected external key K^x is to be used for all P iterations;

an indicator of the selected external key K^x;

an indicator of the selected transformation algorithm A;

a length of a first logical scale of position coding;

an indicator of user information;

the alphabet of transformation AV; and

other transformation algorithm A parameters.

28. The method of claim 2, wherein the information necessary to reverse the conversion process stored in tag data element T may comprise one or more of the following:

an indicator of whether a current iterative step is the P^{th} iteration;

an indicator of whether the selected external key K^x is to be used for all P iterations;

an indicator of the selected external key K^x;

an indicator of the selected transformation algorithm A;
an indicator of user information;

the alphabet of transformation AV;

a length of a first logical scale of position coding;

other transformation algorithm A parameters;

the bit internal identifier K length and shift parameters; and
an indicator of internal identifier K extraction.

29. The method of claim 3, wherein the information necessary to reverse the conversion process stored in tag data element T may comprise one or more of the following:

an indicator of whether a current iterative step is the P^{th} iteration;

an indicator of whether the selected external key K^x is to be used for all P iterations;

an indicator of the selected external key K^x;

an indicator of the selected transformation algorithm A;
an indicator of user information;

the alphabet of transformation AV;

a length of a first logical scale of position coding;

other transformation algorithm *A* parameters;

an indicator of the scrambling function selected;

the bit internal identifier *K* length and shift parameters; and

an indicator of internal identifier *K* extraction.

30. The method of claim 3, wherein the scrambling function is selected from a scrambling matrix comprised of a predefined set of scrambling functions.

31. The method of claim 30, wherein the predefined set of scrambling functions is changed periodically.

32. The method of claim 1, wherein the conversion function further comprises the step of:
determining whether to insert user information into structural data element *S*, and inserting user information into structural data element *S* if determined necessary, thereby providing a means for user authentication and digital signing.

33. The method of claim 2, wherein the conversion function further comprises the step of:
determining whether to insert user information into structural data element *S*, and inserting user information into structural data element *S* if determined necessary, thereby providing a means for user authentication and digital signing.

34. The method of claim 3, wherein the conversion function further comprises the step of:

determining whether to insert user information into structural data element *S*, and inserting user information into structural data element *S* if determined necessary, thereby providing a means for user authentication and digital signing.

35. Computer executable process steps stored on a computer readable medium, the computer executable process steps for structurally converting a binary sequence into an encrypted final image *G*, the computer executable process steps comprising:

forming an image *M* of the binary sequence as a concatenation of a tag data element *T* and structural data element *S*, tag data element *T* comprising information necessary to reverse a conversion process, structural data element *S* comprising a sequence of logical scales of position coding;

selecting a number of conversion steps *P* to be performed;

iteratively executing *P* times a conversion function comprised of the following steps:

selecting a transformation algorithm *A* from a predefined set of transformation algorithms *L*;

selecting an alphabet of transformation *AV* based upon the structural data element *S*;

applying algorithm *A* and alphabet *AV* to structural data element *S* to form a plurality of logical scales of position coding;

forming a transformed structural data element *S'* comprised of a sequence of the logical scales of position coding;

selecting an external key K^x;

forming tag data element *T*;

coding the tag data element *T* with external key K^x to obtain coded tag data element *T''*;

repeating the steps of the conversion function upon a converted image M' comprised of a concatenation of the coded tag data element T'' and the transformed structural data element S';

and forming the encrypted final image G as a concatenation of the coded tag data element T'' and the transformed structural data element S' created upon the P^{th} iteration of the conversion function.

36. Computer executable process steps stored on a computer readable medium, the computer executable process steps for structurally converting a binary sequence into an encrypted final image G, the computer executable process steps comprising:

forming an image M of the binary sequence as a concatenation of a tag data element T and structural data element S, tag data element T comprising information necessary to reverse the conversion process, structural data element S comprising a sequence of logical scales of position coding;

selecting a number of conversion steps P to be performed;

iteratively executing P times a conversion function comprised of the following steps:

selecting a transformation algorithm A from a predefined set of transformation algorithms L;

selecting an alphabet of transformation AV based upon the structural data element S;

applying algorithm A and alphabet AV to structural data element S to form a plurality of logical scales of position coding;

forming a transformed structural data element S' comprised of a sequence of the logical scales of position coding;

stochastically selecting a bit length parameter and a shift parameter which define an internal identifier K within transformed structural data element S';

forming tag data element T;

coding a portion of the tag data element T with internal identifier K to obtain a partially coded tag data element T';

selecting an external key K^x;

coding the partially coded tag data element T' with external key K^x to obtain coded tag data element T'';

stochastically determining whether to extract internal identifier K from transformed structural data element S', and if determined necessary, extracting the internal identifier K from transformed structural data element S' to obtain structural data element S'' and storing internal identifier K in a file of internal identifiers *FID*;

performing the steps of the conversion function upon a converted image M' comprised of a concatenation of the coded tag data element T'' and either transformed structural data element S' if internal identifier K was not extracted, or structural data element S'' if internal identifier K was extracted;

and forming the encrypted final image G as a concatenation of the coded tag data element T'' and either transformed structural data element S' if internal identifier K was not extracted, or structural data element S'' if internal identifier K was extracted.

37. Computer executable process steps stored on a computer readable medium, the computer executable process steps for structurally converting a binary sequence into an encrypted final image G, the computer executable process steps comprising:

forming an image *M* of the binary sequence as a concatenation of a tag data element *T* and structural data element *S*, tag data element *T* comprising information necessary to reverse the conversion process, structural data element *S* comprising a sequence of logical scales of position coding;

selecting a number of conversion steps *P* to be performed;

iteratively executing *P* times a conversion function comprised of the following steps:

selecting a transformation algorithm *A* from a predefined set of transformation algorithms *L*;

selecting an alphabet of transformation *AV* based upon the structural data element *S*;

applying algorithm *A* and alphabet *AV* to structural data element *S* to form a plurality of logical scales of position coding;

forming a transformed structural data element *S'* comprised of a sequence of the logical scales of position coding;

stochastically selecting a bit length parameter and a shift parameter which define an internal identifier *K* within transformed structural data element *S'*;

scrambling internal identifier *K* with a scrambling function to obtain a scrambled internal identifier *K';*

forming tag data element *T*;

coding a portion of the tag data element *T* with scrambled internal identifier *K'* to obtain a partially coded tag data element *T'*;

selecting an external key K^x;

coding the partially coded tag data element *T'* with external key K^x to obtain coded tag data element *T''*;

stochastically determining whether to extract internal identifier *K* from transformed structural data element *S'*, and if determined

necessary, extracting the internal identifier K from transformed structural data element S' to obtain structural data element S'' and storing scrambled internal identifier K' in a file of internal identifiers *FID*;

performing the steps of the conversion function upon a converted image M' comprised of a concatenation of the coded tag data element T'' and either transformed structural data element S' if internal identifier K was not extracted, or structural data element S'' if internal identifier K was extracted;

and forming the encrypted final image G as a concatenation of the coded tag data element T'' and either transformed structural data element S' if internal identifier K was not extracted, or structural data element S'' if internal identifier K was extracted.

38. The computer executable process steps stored on a computer readable medium of claim 35, wherein the external key K^x is selected from a multitude of external keys in an external key file K_{EXT}.

39. The computer executable process steps stored on a computer readable medium of claim 35, wherein the selection of the external key K^x is a stochastic selection.

40. The computer executable process steps stored on a computer readable medium of claim 35, wherein a same external key K^x is selected for use in all iterations.

41. The computer executable process steps stored on a computer readable medium of claim 35, wherein a different external key K^x is selected upon each iteration.

42. The computer executable process steps stored on a computer readable medium of claim 35, wherein the external key K^x is entered by a user during the conversion and reverse conversion process.

43. The computer executable process steps stored on a computer readable medium of claim 35, wherein the selection of transformation algorithm A may be a stochastic selection.

44. The computer executable process steps stored on a computer readable medium of claim 35, wherein the selection of transformation algorithm A may depend upon adherence to a mathematical criterion.

45. The computer executable process steps stored on a computer readable medium of claim 35, wherein the selection of transformation algorithm A may depend upon adherence to a logical criterion.

46. The computer executable process steps stored on a computer readable medium of claim 35, wherein the selection of transformation algorithm A may depend upon adherence to a file size criteria for encrypted final image G.

47. The computer executable process steps stored on a computer readable medium of claim 35, wherein the predefined set of transformation algorithms L may be supplemented.

48. The computer executable process steps stored on a computer readable medium of claim 35, wherein the selection of a number of conversion steps P may be a stochastic selection.

49. The computer executable process steps stored on a computer readable medium of claim 35, wherein the selection of a number of conversion steps P may depend upon adherence to a mathematical criterion.

50. The computer executable process steps stored on a computer readable medium of claim 35, wherein the selection of a number of conversion steps P may depend upon adherence to a logical criterion.

51. The computer executable process steps stored on a computer readable medium of claim 35, wherein the selection of a number of conversion steps P may depend upon adherence to a file size criteria for encrypted final image G.

52. The computer executable process steps stored on a computer readable medium of claim 35, wherein the alphabet of transformation AV is comprised of letters or quants, each letter or quant comprising a segment of structural data element S.

53. The computer executable process steps stored on a computer readable medium of claim 52, wherein a number of bits in each letter or quant is stochastically selected.

54. The computer executable process steps stored on a computer readable medium of claim 52, wherein a number of bits in each letter or quant may depend upon adherence to a mathematical criterion.

55. The computer executable process steps stored on a computer readable medium of claim 52, wherein a number of bits in each letter or quant may depend upon adherence to a logical criterion.

56. The computer executable process steps stored on a computer readable medium of claim 52, wherein a number of bits in each letter or quant may depend upon adherence to a file size criteria for encrypted final image *G*.

57. The computer executable process steps stored on a computer readable medium of claim 35, wherein the information necessary to reverse the conversion process stored in tag data element *T* may comprise one or more of the following:

an indicator of whether a current iterative step is the P^{th} iteration;

an indicator of whether the selected external key K^x is to be used for all *P* iterations;

an indicator of the selected external key K^x;

an indicator of the selected transformation algorithm *A*;

a length of a first logical scale of position coding;

the alphabet of transformation *AV*; and

other transformation algorithm *A* parameters.

58. The computer executable process steps stored on a computer readable medium of claim 36, wherein the information necessary to reverse the conversion process stored in tag data element *T* may comprise one or more of the following:

an indicator of whether a current iterative step is the P^{th} iteration;

an indicator of whether the selected external key K^x is to be used for all *P* iterations;

an indicator of the selected external key K^x;

an indicator of the selected transformation algorithm *A*;

the alphabet of transformation *AV*;

a length of a first logical scale of position coding;

other transformation algorithm *A* parameters;

internal identifier *K* bit length and shift parameters; and

an indicator of internal identifier *K* extraction.

59. The computer executable process steps stored on a computer readable medium of claim 37, wherein the information necessary to reverse the conversion process stored in tag data element T may comprise one or more of the following:

an indicator of whether a current iterative step is the P^{th} iteration;

an indicator of whether the selected external key K^x is to be used for all P iterations;

an indicator of the selected external key K^x;

an indicator of the selected transformation algorithm A;

the alphabet of transformation AV;

a length of a first logical scale of position coding;

other transformation algorithm A parameters;

an indicator of the scrambling function selected;

internal identifier K bit length and shift parameters; and

an indicator of internal identifier K extraction.

60. The computer executable process steps stored on a computer readable medium of claim 37, wherein the scrambling function is selected from a scrambling matrix comprised of a predefined set of scrambling functions.

61. The computer executable process steps stored on a computer readable medium of claim 60, wherein the predefined set of scrambling functions is changed periodically.

62. An apparatus for structurally converting a binary sequence into an encrypted final image G, comprising:

a memory element for storing computer executable process steps;

a processor for executing computer executable process steps;

computer executable process steps comprising:

forming an image M of the binary sequence as a concatenation of a tag data element T and structural data element S, tag data element T comprising information necessary to reverse a conversion process, structural data element S comprising a sequence of logical scales of position coding;

selecting a number of conversion steps P to be performed;

iteratively executing P times a conversion function comprised of the following steps:

selecting a transformation algorithm A from a predefined set of transformation algorithms L;

selecting an alphabet of transformation AV based upon the structural data element S;

applying algorithm A and alphabet AV to structural data element S to form a plurality of logical scales of position coding;

forming a transformed structural data element S' comprised of a sequence of the logical scales of position coding;

selecting an external key K^x;

forming tag data element T;

coding the tag data element T with external key K^x to obtain coded tag data element T'';

repeating the steps of the conversion function upon a converted image M' comprised of a concatenation of the coded tag data element T'' and the transformed structural data element S';

and forming the encrypted final image G as a concatenation of the coded tag data element T'' and the transformed structural data element S' created upon the P^{th} iteration of the conversion function.

63. An apparatus for structurally converting a binary sequence into an encrypted final image G, comprising:

a memory element for storing computer executable process steps;

a processor for executing computer executable process steps;

computer executable process steps comprising:

forming an image M of the binary sequence as a concatenation of a tag data element T and structural data element S, tag data element T comprising information necessary to reverse the conversion process, structural data element S comprising a sequence of logical scales of position coding;

selecting a number of conversion steps P to be performed;

iteratively executing P times a conversion function comprised of the following steps:

selecting a transformation algorithm A from a predefined set of transformation algorithms L;

selecting an alphabet of transformation AV based upon the structural data element S;

applying algorithm A and alphabet AV to structural data element S to form a plurality of logical scales of position coding;

forming a transformed structural data element S' comprised of a sequence of the logical scales of position coding;

stochastically selecting a bit length parameter and a shift parameter

which define an internal identifier K within transformed structural data element S';

forming tag data element T;

coding a portion of the tag data element T with internal identifier K to obtain a partially coded tag data element T';

selecting an external key K^x;

coding the partially coded tag data element T' with external key K^x to obtain coded tag data element T'';

stochastically determining whether to extract internal identifier K from transformed structural data element S', and if determined necessary, extracting the internal identifier K from transformed structural data element S' to obtain structural data element S'' and storing internal identifier K in a file of internal identifiers *FID*;

performing the steps of the conversion function upon a converted image M' comprised of a concatenation of the coded tag data element T'' and either transformed structural data element S' if internal identifier K was not extracted, or structural data element S'' if internal identifier K was extracted;

and forming the encrypted final image G as a concatenation of the coded tag data element T'' and either transformed structural data element S' if internal identifier K was not extracted, or structural data element S'' if internal identifier K was extracted.

64. The apparatus of claim 63, wherein:

the processor is adapted to communicate on a network; and

the computer executable process steps further comprise:

structurally converting the file of internal identifiers *FID* to obtain a converted file of internal identifiers *FID'*, wherein a tag data element formed during the structural conversion of the file of internal identifiers *FID* is coded with an external key selected stochastically from a multitude of external keys in an external key file K_{EXT}; and

transmitting the encrypted final image *G* and structurally converted file of internal identifiers *FID'* to a subscriber or receiver.

65. An apparatus for structurally converting a binary sequence into an encrypted final image *G*, comprising:

a memory element for storing computer executable process steps;

a processor for executing computer executable process steps;

computer executable process steps comprising:

forming an image *M* of the binary sequence as a concatenation of a tag data element *T* and structural data element *S*, tag data element *T* comprising information necessary to reverse the conversion process, structural data element *S* comprising a sequence of logical scales of position coding;

selecting a number of conversion steps *P* to be performed;

iteratively executing *P* times a conversion function comprised of the following steps:

selecting a transformation algorithm A from a predefined set of transformation algorithms L;

selecting an alphabet of transformation AV based upon the structural data element S;

applying algorithm A and alphabet AV to structural data element S to form a plurality of logical scales of position coding;

forming a transformed structural data element S' comprised of a sequence of the logical scales of position coding;

stochastically selecting a bit length parameter and a shift parameter which define an internal identifier K within transformed structural data element S';

scrambling internal identifier K with a scrambling function to obtain a scrambled internal identifier K';

forming tag data element T;

coding a portion of the tag data element T with scrambled internal identifier K' to obtain a partially coded tag data element T';

selecting an external key K^*;

coding the partially coded tag data element T' with external key K^* to obtain coded tag data element T'';

stochastically determining whether to extract internal identifier K from transformed structural data element S', and if determined necessary, extracting the internal identifier K from transformed structural data element S' to obtain structural data element S'' and storing scrambled internal identifier K' in a file of internal identifiers *FID*;

performing the steps of the conversion function upon a converted image M' comprised of a concatenation of the coded tag data element T'' and either transformed structural data element S' if internal identifier K was not extracted, or

structural data element S'' if internal identifier K was extracted;

and forming the encrypted final image G as a concatenation of the coded tag data element T'' and either transformed structural data element S' if internal identifier K was not extracted, or structural data element S'' if internal identifier K was extracted.

66. The apparatus of claim 65, wherein:

the processor is adapted to communicate on a network; and

the computer executable process steps further comprise:

structurally converting the file of internal identifiers *FID* to obtain a converted file of internal identifiers *FID'*, wherein a tag data element formed during the structural conversion of the file of internal identifiers *FID* is coded with an external key selected stochastically from a multitude of external keys in an external key file K_{EXT}; and

transmitting the encrypted final image G and structurally converted file of internal identifiers *FID'* to a subscriber or receiver.

67. The apparatus of claim 62, wherein the external key K^x is selected from a multitude of external keys in an external key file K_{EXT}.

68. The apparatus of claim 62, wherein the selection of external key K^x is a stochastic selection.

69. The apparatus of claim 62, wherein a same external key K^x is selected for use in all iterations.

70. The apparatus of claim 62, wherein a different external key K^x is selected upon each iteration.

71. The apparatus of claim 62, wherein the external key K^x is entered by a user during the conversion and reverse conversion process.

72. The apparatus of claim 67, wherein:

the processor is adapted to communicate on a network; and

the computer executable process steps further comprise:

structurally converting the external key file K_{EXT} to obtain a structurally converted external key file; and

transmitting to a subscriber the structurally converted external key file and an initial key K_{INIT} required to reverse the structural conversion of the structurally converted external key file to obtain the external key file K_{EXT}.

73. The apparatus of claim 62, wherein the selection of transformation algorithm A may be a stochastic selection.

74. The apparatus of claim 62, wherein the selection of transformation algorithm A may depend upon adherence to a mathematical criterion.

75. The apparatus of claim 62, wherein the selection of transformation algorithm A may depend upon adherence to a logical criterion.

76. The apparatus of claim 62, wherein the selection of transformation algorithm A may depend upon adherence to a file size criteria for encrypted final image G.

77. The apparatus of claim 62, wherein the predefined set of transformation algorithms L may be supplemented.

78. The apparatus of claim 62, wherein the selection of a number of conversion steps P may be a stochastic selection.

79. The apparatus of claim 62, wherein the selection of a number of conversion steps P may depend upon adherence to a mathematical criterion.

80. The apparatus of claim 62, wherein the selection of a number of conversion steps P may depend upon adherence to a logical criterion.

81. The apparatus of claim 62, wherein the selection of a number of conversion steps P may depend upon adherence to a file size criteria for encrypted final image G.

82. The apparatus of claim 62, wherein the alphabet of transformation AV is comprised of letters or quants, each letter or quant comprising a segment of structural data element S.

83. The apparatus of claim 82, wherein a number of bits in each letter or quant is stochastically selected.

84. The apparatus of claim 82, wherein a number of bits in each letter or quant may depend upon adherence to a mathematical criterion.

85. The apparatus of claim 82, wherein a number of bits in each letter or quant may depend upon adherence to a logical criterion.

86. The apparatus of claim 82, wherein a number of bits in each letter or quant may depend upon adherence to a file size criteria for encrypted final image G.

87. The apparatus of claim 62, wherein the information necessary to reverse the conversion process stored in tag data element T may comprise one or more of the following:

an indicator of whether a current iterative step is the P^{th} iteration;

an indicator of whether the selected external key K^x is to be used for all P iterations;

an indicator of the selected external key K^x;

an indicator of the selected transformation algorithm A;

a length of a first logical scale of position coding;

the alphabet of transformation AV; and

other transformation algorithm A parameters.

88. The apparatus of claim 63, wherein the information necessary to reverse the conversion process stored in tag data element T may comprise one or more of the following:

an indicator of whether a current iterative step is the P^{th} iteration;

an indicator of whether the selected external key K^x is to be used for all P iterations;

an indicator of the selected external key K^x;

an indicator of the selected transformation algorithm A;

the alphabet of transformation AV;

a length of a first logical scale of position coding;

other transformation algorithm A parameters;

the bit internal identifier K length and shift parameters; and

an indicator of internal identifier K extraction.

89. The apparatus of claim 63, wherein the information necessary to reverse the conversion process stored in tag data element T may comprise one or more of the following:

an indicator of whether a current iterative step is the P^{th} iteration;

an indicator of whether the selected external key K^x is to be used for all P iterations;

an indicator of the selected external key K^x;

an indicator of the selected transformation algorithm A;

the alphabet of transformation AV;

a length of a first logical scale of position coding;

other transformation algorithm A parameters;

an indicator of the scrambling function selected;

the bit internal identifier K length and shift parameters; and

an indicator of internal identifier K extraction.

90. The apparatus of claim 65, wherein the scrambling function is selected from a scrambling matrix comprised of a predefined set of scrambling functions.

91. The apparatus of claim 90, wherein the predefined set of scrambling functions is changed periodically.